

Railroad Simulation Using OpenTrack

Andrew Nash & Daniel Huerlimann
*Swiss Federal Institute of Technology
Institute for Transportation Planning and Systems
Switzerland*

Abstract

OpenTrack is a user-friendly railroad network simulation program developed at the Swiss Federal Institute of Technology's Institute for Transportation Planning and Systems (ETH IVT). It is a microscopic model that simulates rail system operations based on user defined train, infrastructure, and timetable databases. OpenTrack functions as a railroad laboratory, by, for example, allowing users to define incidents and take infrastructure out of service to evaluate alternative scenarios. The program uses a mixed discrete/continuous simulation process that calculates both the continuous solution of train motion equations and the discrete processes of signal box states and delay distributions. It generates a wide variety data that can be easily presented in many formats including graphs (e.g. time-space diagrams), tables, and images. OpenTrack's main uses have been to evaluate and test infrastructure plans and operating schedules to optimize network and timetable design. It can be run on several different computer platforms and incorporates the benefits of object oriented programming language with a common data interface structure.

Keywords: railway simulation, railway planning, railway operations analysis, RailML, XML, object oriented modeling.

1 Introduction

Railroad planning is particularly challenging because different improvements can be used to achieve project objectives; improvements can be divided into three general categories: infrastructure, rolling stock, and operations. Improvements in each category need to be evaluated against improvements in other categories to develop the optimal investment plan.

A good example of a structured approach to trading off different types of improvements in these categories is the Swiss National Railroad's (SBB) Integrated Product Planning Process. [1] The SBB views this process as a Planning Triangle with three elements at the corners: Products, Rolling Stock, and Infrastructure (see Figure 1). Products are the services and schedules operated (e.g. commuter rail, intercity rail, freight); rolling stock means the type of rolling stock used to provide a particular service; and infrastructure consists of the physical system (e.g. tracks, signal systems, stations). SBB planners use iterative techniques to evaluate changes in each of these elements to optimize the system as a whole. This triangular depiction effectively communicates the relationship between the three elements and their ability to meet market demand.

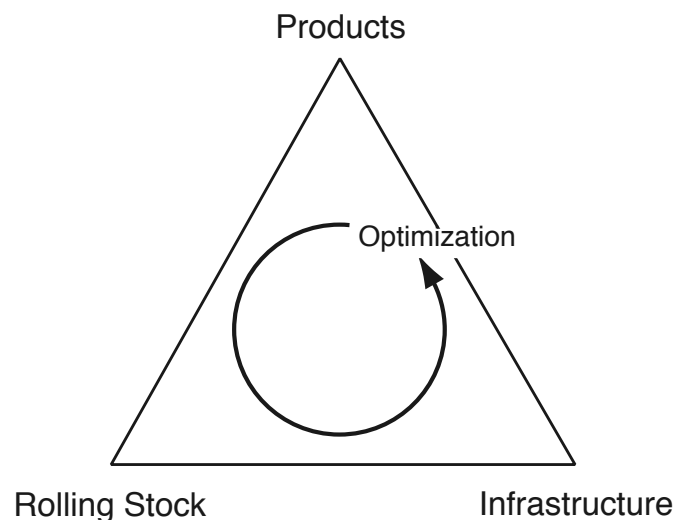


Figure 1. Swiss National Railways' planning triangle.

In an example of this process, the SBB decided to use tilting trains to provide high speed service (a rolling stock solution) rather than fully rebuilding tracks on a particular corridor (an infrastructure solution) because the former was found to be more cost effective.

Computer simulation is a particularly important and useful tool for evaluating different railroad improvement strategies for the following reasons:

- **Understanding Capacity** – Railway capacity is not intuitively obvious. Even lines with very little service may be operating at capacity.
- **Highly Interrelated Infrastructure** – An infrastructure improvement in one location can have significant impacts in another location, sometimes far from the improvement. A rail simulation program can identify the impacts of such changes throughout the modeled network.
- **High Cost of Rail Infrastructure** – Improving a railroad is expensive, not only are the physical improvements costly, but costs for taking a line out of service during construction and for additional right-of-way can be significant. Furthermore, a poorly planned improvement will increase the railroad's long term operating costs and problems.

Given these factors, many rail-planning experts recommend completing as much modeling as possible before starting a railroad improvement program. In general, the more modeling done up front, the less expensive the overall project will be, since modeling enables the plan to be refined to its most essential elements. [2]

The first step in using computer models in the railroad planning is to calibrate the base case model. This should accurately replicate observed railroad operations with the existing infrastructure, rolling stock, and schedules. Once the model has been calibrated it can be used to investigate many issues including estimating the stability of new timetables, determining the minimum infrastructure requirements for a given timetable, or evaluating the impact of rolling stock changes. A significant benefit of models is their ability to evaluate the impact of incidents or time-based network changes (e.g. maintenance) on railroad operations.

Computer simulation is especially valuable for railroad planning since, once developed and calibrated, models can be used to compare the benefits, impacts, and costs of various different improvement packages. To analyze more than a few improvement packages by hand would be prohibitively time consuming. Thus effective railroad simulation models enable planners to identify and evaluate more alternatives ultimately leading to more creative and comprehensive problem solutions.

While computer simulation is an excellent tool for analysis and planning of railroads, railroad network simulation programs have the following limitations:

- Programs must be validated to actual conditions.
- Yard operations must be modeled separately.
- Resource constraints such as crew scheduling are largely ignored (although some specialized software does address resource constraints).
- Simulations only include the modeled study area.
- Simplifying assumptions generally create an inherent optimism about overall congestion, schedule adherence, and recoverability. [3]

Given these limitations, especially the last one, it is critical that all simulation results be carefully reviewed and discussed with those familiar with operations. There is no substitute for real experience in the planning process.

2 Types of Rail Simulation Programs

At its most basic level a rail simulation program is a set of simplifying mathematical assumptions that attempt to duplicate actual train operations. It allows users to evaluate the impacts of user-defined changes on the rail system operation.

There are many types of railroad simulation programs; each of which is designed to answer a different type of question. For example there are models designed to help dispatchers decide how to route trains through a network in real time. Similarly there are models designed to help planners identify infrastructure requirements for planned schedules ten years in the future. As with most

planning processes, a critical component of railroad planning is choosing the best computer simulation tool to use in any given situation.

Rail simulation models can be categorized as macroscopic or microscopic similar to other transportation models. Both types are outlined below.

2.1 Macroscopic Simulation Models

Macroscopic models use average or other statistical data to evaluate operation of the transportation system; they do not model individual unit (e.g. train) operations nor do they consider how trains are impacted by other trains or how safety systems impact train performance. A common type of macroscopic model is the NEMO model developed at the University of Hannover.

2.2 Microscopic Simulation Models

Microscopic simulation models attempt to replicate the actual operation of a railroad over time. They do this by modeling the operation of each individual train during a user-defined time step (often one second) and then repeating the process for the entire simulation period.

Microscopic models consider the impact of trains on each other when they simulate train operations. For example, if during a particular time step, a train occupies a block that a second train wishes to occupy, then the program will model the second train as performing according to the (user defined) safety system parameters. In the case of a simple block signal system the second train would stop at the block entry signal and wait until the first train clears the block before proceeding.

There are two types of microscopic models: synchronous and asynchronous. Synchronous models simulate all train operations in a single model run while asynchronous models simulate operations in a series of model runs.

In an asynchronous simulation the highest priority trains are modeled in the first run, then this schedule is locked and the second set of trains is modeled; the operation of the second set of trains does not impact the first set of trains, but is impacted by the first set (similarly, the operation of the third set of trains does not impact the first or second sets, and so forth). These types of models are often used for timetable construction since they can replicate ideal operational planning. A good example of this type of program is STRESI developed by RWTH in Aachen.

In contrast, synchronous models simulate all the trains operating in the modeled network at the same time, thus they provide a good way to simulate realistic operating situations (e.g. the impact of train delays on network operations). These models enable users to specify rules that the program uses to make dispatching decisions when there are conflicts between trains. These rules include simple train priorities (e.g. passenger trains before freight trains) as well as more complex rules designed to optimize some particular function (e.g.

dynamic overtaking). In essence these models attempt to replicate good dispatching decisions.

OpenTrack is a microscopic synchronous railroad simulation model. It provides users with a great deal of flexibility for defining different dispatching logic as well as operational variables in a user-friendly manner.

3 Object Oriented Programming and XML

Simulation is one of the most valuable tools available for railroad planning and design since it allows planners to evaluate different infrastructure, rolling stock, and timetable alternatives before implementation. In general the more simulation the better, and therefore using the most efficient rail simulation programs improves railroad planning. Two computer programming innovations that have helped increase the effectiveness of modern rail simulation programs are object oriented programming and XML-based data formats.

An object oriented computer program is created by combining modules of standardized computer code with application-specific algorithms. RailML is an open source XML-based language developed for railroad applications. [4] It is a simple and efficient way to transfer data between programs used to model different aspects of railroad operations. Programs using RailML produce export files with the RailML structure that can be used directly by other programs.

OpenTrack makes use of both object oriented programming and the RailML data structure, providing rail planners with benefits including:

- **Efficiently use multiple application programs** – Different rail simulation programs are useful for answering different questions, using the RailML structure allows users to mix and match the best tools for the given task because data transfer is fast and efficient.
- **Quickly modify application programs** – Object oriented programming allows developers to quickly modify and customize programs to meet user needs.
- **User-friendliness** – Object oriented programming and the RailML structure allow users to enter data and begin using programs quickly.
- **Ability to replicate existing output** – Object oriented programming enables developers to easily replicate output from existing programs, simplifying the transition from one program to another for users.

Together these benefits enable users to perform more modeling and more directly applicable modeling than they could using independent computer simulation models.

4 OpenTrack Rail Simulation Program

OpenTrack was developed at the Swiss Federal Institute of Technology's Institute for Transportation Planning and Systems (ETH IVT). The project's goal was development of a user-friendly railroad simulation program that can run on different computer platforms and can answer many different questions about

railway operations. [5] Figure 2 illustrates the three main elements of OpenTrack: data input, simulation, and output.

OpenTrack is a microscopic synchronous railroad simulation model. As such it simulates the behaviour of all railway elements (infrastructure network, rolling stock, and timetable) as well as all the processes between them. It can be easily used for many different types of projects including testing the stability of a new timetable, evaluating the benefits of different long-term infrastructure improvement programs, and analyzing the impacts of different rolling stock.

4.1 Input Data

OpenTrack administers input data in three modules: rolling stock (trains), infrastructure, and timetable. Users enter input information into these modules and OpenTrack stores it in a database structure. Once data has been entered into the program, it can be used in many different simulation projects. For example, once a certain locomotive type has been entered into the database, that locomotive can be used in any simulation performed with OpenTrack. Similarly, different segments of the infrastructure network can be entered separately into the database and then used individually to model operations on the particular segment or together to model larger networks.

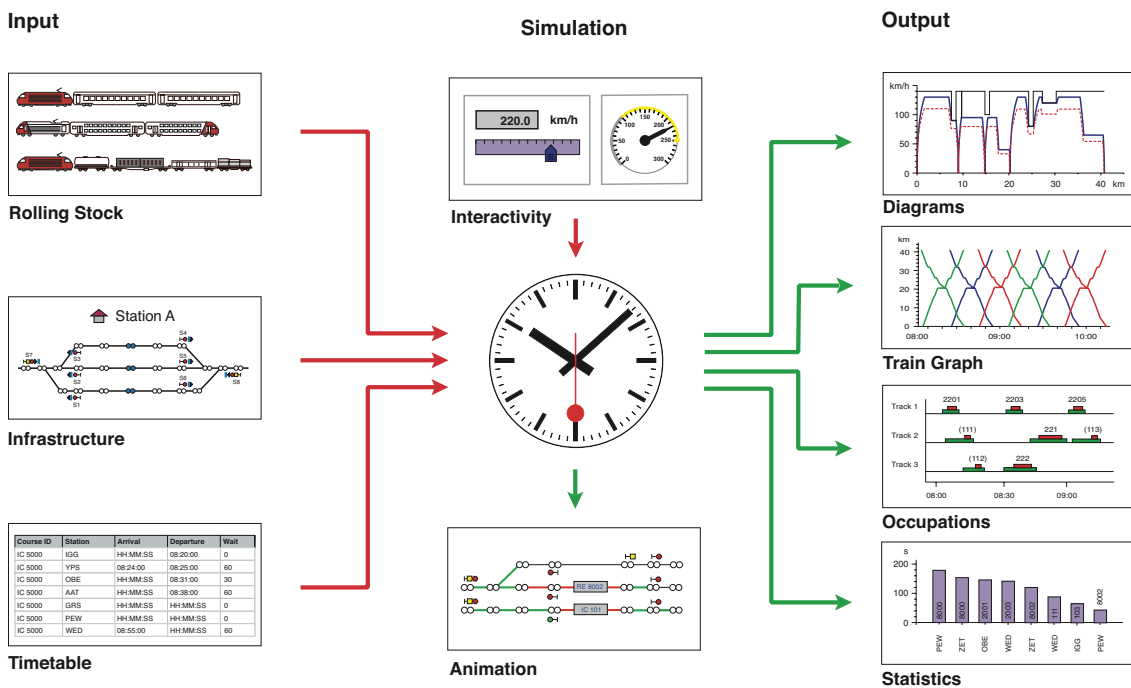


Figure 2. Main elements of OpenTrack.

Train data (locomotive and wagons) is entered into the OpenTrack database with easy to use forms displayed using pull down menus. Infrastructure data (e.g. track layout, signal type/location) is entered with a user-friendly graphical interface; quantitative infrastructure data (e.g. elevation) is added using input forms linked to the graphical elements. Following completion of the RailML data structure for rolling stock and infrastructure, OpenTrack will be modified to enable train and infrastructure data to be directly imported from RailML data files. Figure 3 presents a typical OpenTrack document.

Timetable data is entered into the OpenTrack database using forms. These forms include shortcuts that enable data input to be completed efficiently. For example, users can designate hourly trains that follow the same station stopping pattern an hour later. Since OpenTrack uses the RailML structure for timetable data, timetable data can also be entered directly from various different program output files as well as database files.

One advantage of OpenTrack is that it enables users to adjust many variables that impact railroad operations. For example, users can simulate the impact of weather on traction by specifying the adhesion scenario (good, normal, bad). OpenTrack then estimates locomotive traction power using a percentage (also user-defined) of that calculated using the Curtius and Kniffler formula. [6] While OpenTrack provides standard default values for all variables, having the ability to adjust variables makes the program quite useful.

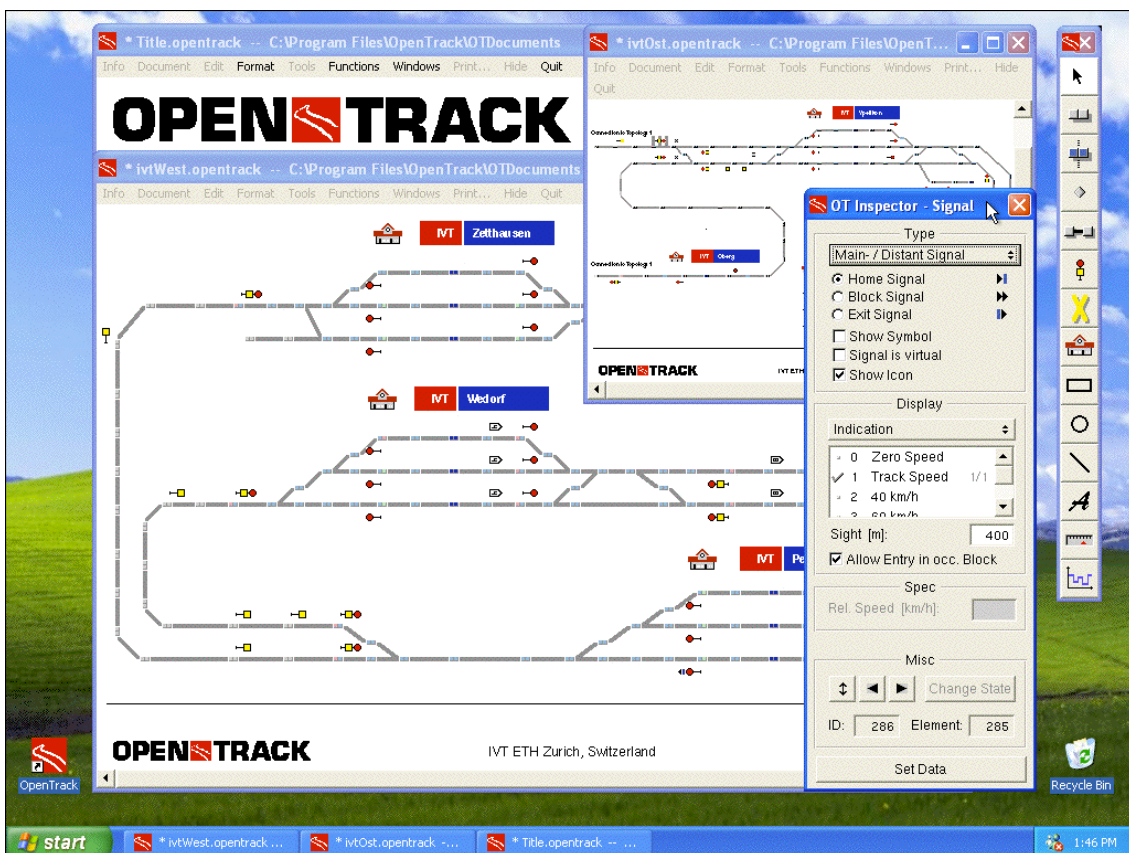


Figure 3. OpenTrack Document.

4.2 OpenTrack Simulation Process

In order to run a simulation using OpenTrack the user specifies the trains, infrastructure and timetable to be modeled along with a series of simulation parameters (e.g. animation formats) on a preferences window. During the simulation, OpenTrack attempts to meet the user-defined timetable on the specified infrastructure network based on the train characteristics. OpenTrack uses a mixed continuous/discrete simulation process that allows a time driven running of all the continuous and discrete processes (of both the vehicles and the safety systems) under the conditions of the integrated dispatching rules.

The continuous simulation is dynamic calculation of train movements based on Newton's motion formulas. For each time step, the maximum force between the locomotive's wheels and the tracks is calculated and then used to calculate acceleration. Next, the acceleration function is integrated to provide the train's speed function and is integrated a second time to provide the train's position function. [6]

The discrete simulation process models operation of the safety systems; in other words, train movements are governed by the track network's signals. Therefore, parameters including occupied track sections, signal switching times, and restrictive signal states all influence the train performance. OpenTrack supports traditional multi-aspect signaling systems as well as new moving block train control systems (e.g. European Train Control System – ETCS signaling).

4.3 Dynamic Rail Simulation

OpenTrack is a dynamic rail simulation program. As such, the simulated operation of trains depends on the state of the system at each step in the process as well as the original user-defined objective data (e.g. desired schedule).

A simple way of describing dynamic rail simulation is that the program decides what routes trains use while the program is running. For example, when building the network, users identify various different routes that trains can use between two points; OpenTrack decides, during the simulation, which route the train will use by assigning the train the highest priority route available. If the first priority is not available, OpenTrack will assign the train the second highest priority route and so on.

OpenTrack's dynamic nature allows users to assign certain attributes to specified times in the simulation. Thus, users can assign a delay to a particular train at a given station and time, rather than being limited to assigning a delay at the start and using it through the entire simulation. Similarly, users can define other types of incidents (e.g. infrastructure failures, rolling stock breakdowns) for particular times and places.

Finally, dynamic simulation enables users to run OpenTrack in a step-by-step process and monitor results at each step. Users can also specify exactly what results are displayed on the screen. Running OpenTrack in a step-by-step mode with real time data presented on screen helps users to identify problems and develop alternative solutions.

4.4 OpenTrack Output

One of the major benefits of using an object oriented language is the great variety of data types, presentation formats, and specifications that are available to the user. During the OpenTrack simulation each train feeds a virtual tachograph (output database), which stores data such as acceleration, speed, and distance covered. Storing the data in this way allows users to perform various different evaluations after the simulation has been completed.

OpenTrack allows users to present output data in many different formats including various forms of graphs (e.g. time-space diagrams), tables, and images. Similarly, users can choose to model the entire network or selected parts, depending on their needs. Output can be used either to document a particular simulation scenario or as an interim product designed to help users identify input modifications for another model run. Figure 4 presents a graphical timetable (train graph) and Figure 5 presents a speed-distance diagram, two of several types of output that are possible using OpenTrack. [7]

5 Conclusions

OpenTrack is an efficient and effective railroad simulation program. It has been used successfully in many different railway planning projects throughout the world. The program's use of object oriented programming and the RailML data structure makes it particularly effective since the program can be modified relatively easily to address specific applications and since data can be transferred easily to and from other programs based on RailML.

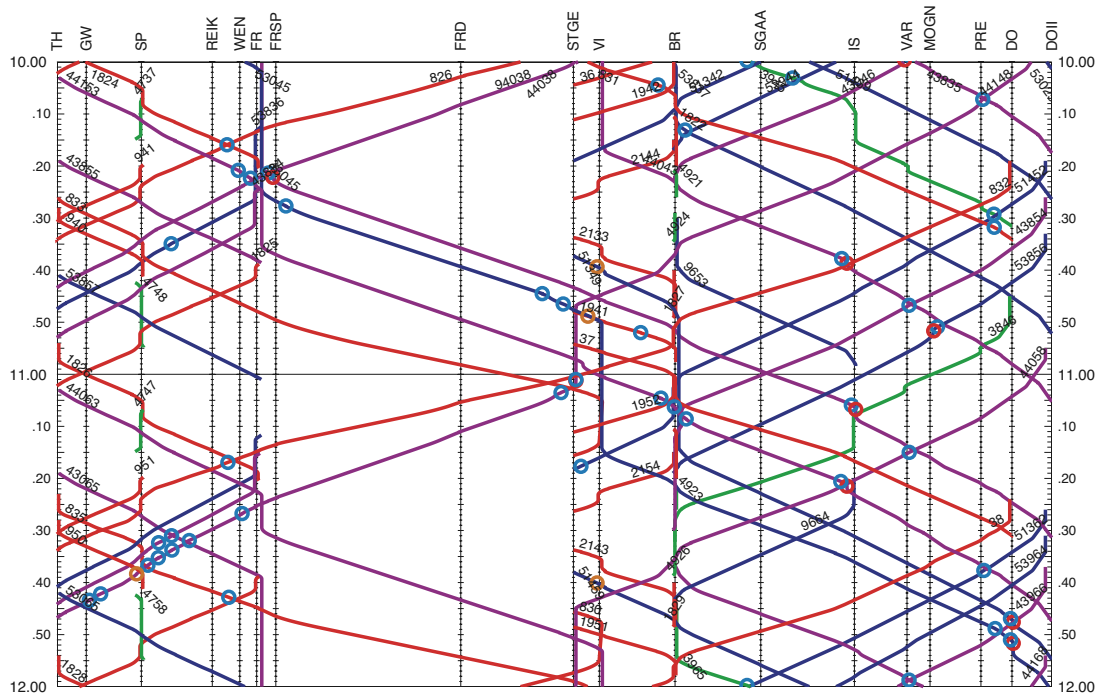


Figure 4. OpenTrack Output: Train Graph

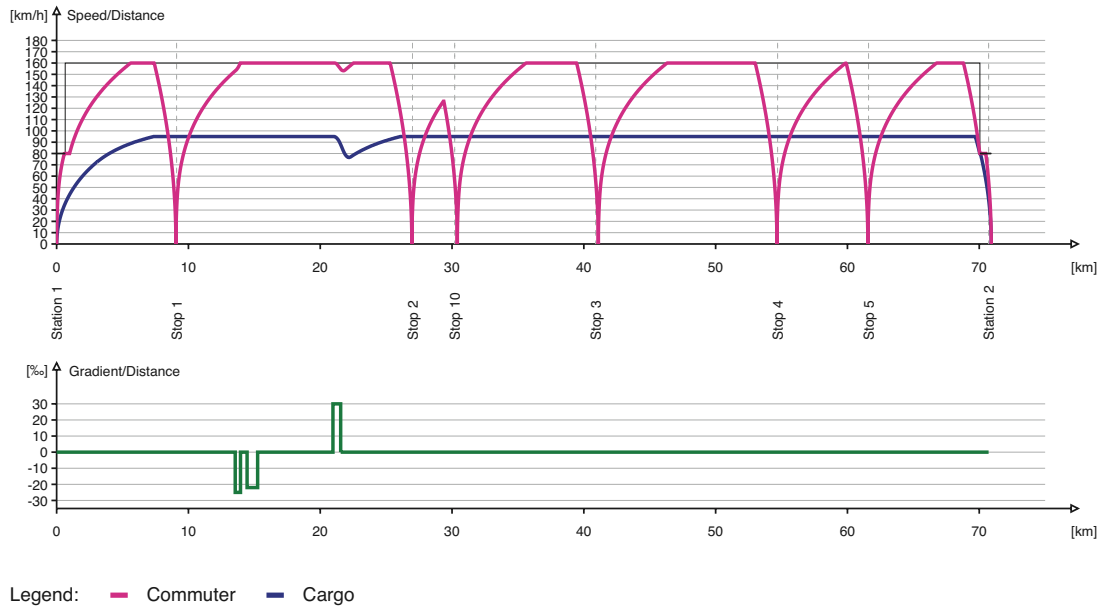


Figure 5. OpenTrack Output: Speed-Distance Diagram.

References

- [1] Grossenbacher, Peter. "Service Planning Strategies: The Example of the Passenger Transport Department of the Swiss Federal Railways." *Rail International*, June/July 2001, 118.
- [2] Nash, Andrew; Best Practices in Shared-Use Rail Operations; Mineta Transportation Institute, San Jose State University; San Jose, California; June 2003.
- [3] Gibson, John; Train Performance Calculators and Simulation Models; Handout, Transportation Research Board, "TRB Workshop on Railroad Capacity and Corridor Planning." January 13, 2002.
- [4] For more information on RailML see the project website: www.railml.org
- [5] Huerlimann, Daniel; Object oriented modeling in railways; ETH Dissertation Nr. 14281; 2001 (in German).
- [6] Huerlimann, Daniel and Andrew Nash; OpenTrack – Simulation of Railway Networks, User Manual Version 1.3; ETH Zurich, Institute for Transportation Planning and Systems; May 2003; Page 58.
- [7] For more information on OpenTrack see the project website: www.opentrack.ch